IN THE SPECIFICATION:

On page 2, beginning at line 21, please amend the specification as follows:

FIG. 1 is a block diagram of a processor cache according to an embodiment of the present invention. According to the embodiment, the cache 100 may be populated into a plurality of cache entries 110. The embodiment of FIG. 1 illustrates the cache entries 110 organized into sets and ways. As shown in FIG. 1, way 0 may include a predetermined number N of cache entries. Each cache entry 110 may be organized into a tag field 120, a data field 130 and a ~~stage~~ state field (labeled "S"). Data to be stored in the cache 100 may be stored in the data fields 130 of the cache entries 110. Partial address information identifying a source of the data stored in a data field 130 may be stored in the tag field 120. Cache coherency state information may be stored in the state field S. For simplicity's sake, the architecture of a single way (way 0) is illustrated in FIG. 1; the architecture of way 0 may extend to the other ways 1, 2, etc. In practice a cache may include fewer or more ways than the four ways shown in FIG. 1.

On page 7, beginning with line 17, please amend the specification as follows:

If the data request is a snoop probe, the cache manager 180 may generate a TI UOP 370. The TI UOP 370 may cause data to be read from the tag fields 120 and state fields S from addressed cache entries 110 in all ways of the cache 100. The data fields 130 throughout the cache 100 and the victim allocation unit 160 may be powered down during the TI UOP 370. Following the TI UOP 370, if a way comparator 150 indicates a tag match, the cache manager may generate a TW UOP 380 if the state information from the matching cache entry 110 indicate that the data were held in exclusive state ("E state" in the drawings) or if the snoop probe were not a Go-to-Shared snoop ("G2S" in the drawings). The TW UOP 380 may update the data held in the state field S of the matching way. All other ways and the victim allocation unit 160 may be powered down during processing of the TW UOP 380. Thereafter, the cache manager 180 may return to the idle state 300. Following the TI UOP 370, if the state information from the matching cache entry 110 indicates that the data is invalid or modified, the cache manager 180 may return to the idle state 300.

On page 9, beginning with line 5, please amend the specification as follows:

When the data request is a read operation, the cache manager 180 may generate an RLU UOP 410. The RLU UOP 410 may energize the tag fields 120, data fields 130 and state fields S of all ways in the cache 100 (FIG. 1) as well as the LRU 170 in the set identified by the $ADDR_{set}$ so that data may be read from them. A cache hit/miss decision may be made from tag comparisons in each of the ways. If a cache miss occurs, the cache manager 180 may determine, from the state information, ~~of~~ a victim way (identified by the LRU 170). If the victim way is dirty, the cache manager 180 may generate an Evict UOP 420 to evict the dirty data from the cache 100. The Evict UOP 420 may cause only one of the ways to be energized; all other ways and the victim allocation unit 160 may be powered down. Thereafter, or if the victim data was not dirty, the cache manager 180 may return to the idle state 400.